ERDC MSRC/PET TR/00-13

Error and Shape Quality Indicators for Adaptive Refinement and Deforming Finite Elements

by

G. F. Carey
J. T. Oden
A. K. Patra
A. I. Pehlivanov
S. Prudhomme
D. Littlefield

30 March 2000

07h0022000

Error and Shape Quality Indicators for Adaptive Refinement and Deforming Finite Elements

G. F. Carey[*], J. T. Oden[*], A. K. Patra[†],
A. I. Pehlivanov[*], S. Prudhomme[*] and D. Littlefield[‡]

March 30, 2000

Abstract

We examine the problem of error control and element shape quality maintenance during adaptive refinement/coarsening simulations and with deforming finite elements. Of particular interest are large deformation Lagrangian simulations of impact but the ideas are more broadly applicable. Two key questions are considered: (1) the control of the error in the numerical solution by means of adaptive refinement and coarsening guided by appropriate error indicators, and (2) the control of shape degradation of the elements as the grid deforms. In both Eulerian and Lagrangian simulations, element size and shape can seriously impact the timestep size. Indicators for error and shape control are discussed and the ideas are tested in exploratory simulations for impact problems using representative explicit codes.

1 Introduction

There have been numerous theoretical and algorithmic studies directed to the various aspects of mesh optimization in finite element analysis. These include development of error indicators in order to guide mesh refinement/coarsening, algorithms for adding points, subdividing elements, faces or edges, as well as studies involving the underlying data structure. Also, there is an interest in node redistribution and moving grid techniques, particularly for problems with evolving domains. Indeed, the possible distortion of elements imply several associated problems such as mesh tangling and element degeneracy, which may affect accuracy or simply cause failure of the numerical scheme. More recently these ideas have been merging since algorithms that support simultaneous refinement and coarsening can be viewed

[*]Texas Institute for Computational and Applied Mathematics (TICAM), University of Texas at Austin, Austin, TX 78712

[†]Department of Mechanical Engineering, SUNY at Buffalo, Buffalo, NY 14260

[‡]Institute for Advanced Technology, Austin, TX

as a type of redistribution strategy, the main distinction having to do with the supporting data structure approaches. For an overview of some of these aspects, we refer to the comprehensive discussion in [15]. For details on grid generation, see for example [20, 26, 44, 46] and on analysis of error indicators, see [1, 3–7, 18, 22, 27, 28].

One issue, which is increasingly raised, is how to assess the quality of a given mesh, especially when performing automatic adaptive refinement. Mesh adaption schemes may produce slender elements, with small and large angles, particularly in the case of moving grids. Refinement to smaller elements and element degeneracy usually alters the performance of time-discretization schemes. In order to satisfy stability conditions in the explicit schemes considered later the integration timestep may become extremely small as elements approach a degenerate sliver or point. This may be a problem even in regions where the solution gradients are not so large, so accuracy may not be an issue, although such elements are not recommended in practice for reasons of ill conditioning. In the present work, we consider the possibility of controlling both the approximation error and the element shape quality.

A number of mesh quality indicators have been suggested in the literature, mainly on the basis of intuitive geometric reasoning. There are also approaches that appeal more to the mapping properties and the Jacobian of the transformations between the reference element and the element in the physical domain. These actually relate to the geometry itself since the integral of the Jacobian determinant on the element provides a measure of the element volume. Also, we will show that the mapping from the reference element and its inverse arise naturally from interpolation theory in the *a priori* error estimates of finite element approximations, and, in our opinion, this latter argument provides a logical basis for developing suitable classes of *element quality indicators*. This is the starting point in the present work and we remark that this approach has also been undertaken in a recent work [25] where the condition number of the Jacobian matrix is studied as a quality indicator to be used in various *objective functions* for grid smoothing.

One other objective in this project is to explore the applicability of different types of error indicators when access to the software kernels or the physics models is rather limited. We have in mind here *legacy codes* or *large proprietary software systems*, for which stand alone modules can only be interfaced with the proprietary code. More specifically, an infrastructure capable of supporting dynamic creation and deletion of elements and nodes as well accuracy or reliability assessment is required. The test codes for our work described later are the Lagrangian EPIC code (Elasto-Plastic Impact Calculations) and the Eulerian CTH code, which are representative of codes for this class of impact problems. While the literature on computational impact simulations is rather vast [2, 11], the use of solution adaptive meshing for these problems is relatively recent [14, 19, 36].

A major goal of this research activity is to develop an efficient adaptive method which would dramatically improve simulations of impact but the ideas are obviously of more general interest. Central to our success is the use of effective refinement and shape quality indicators with adaption strategies, as well as the development of a supporting software infrastructure. One of the developments arising from our exploratory studies is the design of *hybrid error indicators* which may be used to control both the numerical error and element shape quality.

Our treatment of the error due to mesh distortion as the geometry changes makes the algorithm much more robust for Lagrangian computations for high velocity impact with EPIC. Likewise, in our Eulerian simulations with CTH, we consider a restricted class of adaptive refinement/coarsening strategies based on 2:1 block subdomain refinement.

The outline of the report is as follows. In Section 2, we discuss basic interpolation estimates, a residual error estimate for an explicit time discretization scheme, a representative flux jump indicator and a flux/stress post-processing recovery indicator. In Section 3, we consider the question of shape quality error indicators both for Eulerian and Lagrangian calculations as well as transient applications with general remarks on the hybrid indicators and software issues. Next, in Section 4, we investigate additive and multiplicative hybrid indicators that are tested in impact simulations. We emphasize that the discussion here is not meant to be comprehensive. Instead it describes some representative ideas and exploratory simulations of impact using these concepts.

2 Error Estimation and Error Indicators

Solution adaptive meshing implies the selection of adequate error measures so as to properly assess the quality of numerical approximations. A satisfactory measure is essential for constructing new meshes that will better meet the goals of the analyst. There has been a large amount of research on developing such error estimators (e.g. see [1, 29, 47]). However, relatively few studies have been directed towards error estimation for transient non-linear problems like the impact problem considered here. Of particular interest are estimators that could be easily integrated into the explicit time integration schemes widely used in nonlinear transient impact simulations. We recall that the other objective is to design a criterion to avoid creating ill-shaped elements as the mesh is repeatedly refined or deforms.

There exist numerous studies of error indicators for adaptive element refinement or node redistribution. The earliest work used the *element residuals* as error indicators [3–5]. Later works focussed on *interpolation estimates* [18] or on *residual estimates* based on solving local boundary value problems [1, 6, 7, 22, 27, 28], not to mention the *patch-recovery technique* involving superconvergence properties of the finite element discretizations [43, 51–53]. A more recent approach is the so-called *goal-oriented error estimation and adaptation* [10, 33, 37, 38, 41, 42, 49] which controls the error in terms of *quantities of interest*.

One major issue in practical applications involving either *legacy codes* or *large scale proprietary systems* is the fact that there may be limited access to the software kernels. Nevertheless, it is usually possible to add a new module that would compute an error indicator only using the computed solution and the corresponding mesh at a given time step. However, residual type error indicators generally necessitate more detailed knowledge of the constitutive models and physics, as well as the numerical schemes or algorithms being implemented. For these reasons, we will focus in this project on less invasive *recovery* indicators.

We first introduce some classical interpolation error estimates as they will be employed several times in this report. We then present a methodology to compute error estimates

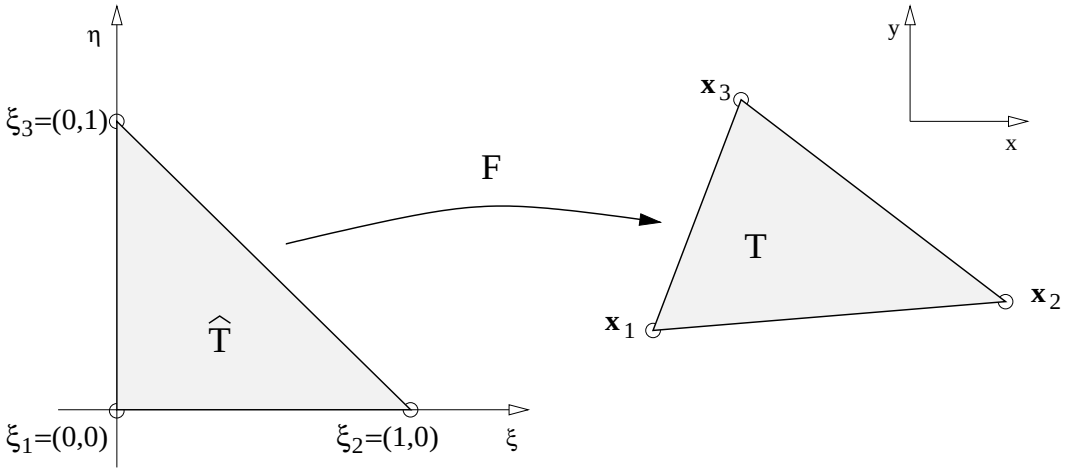

Figure 1: Reference triangle $\widehat{T}$ and its image T in the physical domain.

based on residuals for a time-dependent model problem. The results of the analysis will justify the use of flux-jump based or patch-recovery indicators reviewed at the end of this section.

2.1 Interpolation Estimates

For convenience, let us consider here the case of an arbitrary triangle in the mesh which is mapped by an affine mapping from the reference right isosceles triangle [9]:

$$\widehat{T} = \{\boldsymbol{\xi} = (\xi, \eta) : \xi \geq 0, \eta \geq 0, \text{ and } 1 - \xi - \eta \geq 0\}. \tag{2.1}$$

The affine map is defined as:

$$F : \boldsymbol{\xi} \in \widehat{T} \longrightarrow \boldsymbol{x} = B\boldsymbol{\xi} + \boldsymbol{x}_0 \in T$$

such that

$$F(\boldsymbol{\xi}_1) = \boldsymbol{x}_1, \qquad F(\boldsymbol{\xi}_2) = \boldsymbol{x}_2, \qquad F(\boldsymbol{\xi}_3) = \boldsymbol{x}_3.$$

where $\boldsymbol{\xi}_i$ and $\boldsymbol{x}_i$, $i = 1, 2, 3$, denote the coordinates of the vertices on the reference element $\widehat{T}$ and on its image T respectively (see Fig. 1). We note that the entries of the matrix B are actually given by

$$B = \begin{bmatrix} (x_2 - x_1) & (x_3 - x_1) \\ (y_2 - y_1) & (y_3 - y_1) \end{bmatrix}. \tag{2.2}$$

Let v be an arbitrary function defined on T and $\mathcal{I}$ the standard nodal interpolation operator; i.e. $\mathcal{I}v$ is linear and

$$\mathcal{I}v(\boldsymbol{x}_1) = v(\boldsymbol{x}_1), \qquad \mathcal{I}v(\boldsymbol{x}_2) = v(\boldsymbol{x}_2), \qquad \mathcal{I}v(\boldsymbol{x}_3) = v(\boldsymbol{x}_3).$$

Using standard definitions of the norms [34], we now state classical lemmas on properties of the interpolation error [13, 21]:

Lemma 2.1 *Let F be the bijective mapping $F : \widehat{T} \to T$ such that $F(\widehat{\boldsymbol{x}}) = B\widehat{\boldsymbol{x}} + \boldsymbol{x}_0$ where B is non-singular. If $v \in H^r(T)$, $r \geq 0$, then $\widehat{v} = v \circ F \in H^r(\widehat{T})$ and there exists a positive constant $C = C(\widehat{T}, r)$ such that*

$$|v|_{r,T} \leq C \, \|B^{-1}\|^r \, |\det B|^{1/2} \, |\widehat{v}|_{r,\widehat{T}}, \tag{2.3}$$

$$|\widehat{v}|_{r,\widehat{T}} \leq C \;\; \|B\|^r \, |\det B|^{-1/2} \, |v|_{r,T}. \tag{2.4}$$

Lemma 2.2 *Let $\widehat{\mathcal{I}}$ be the interpolation operator on the reference element. For $m \geq 2$, there exists a positive constant $C = C(r, m)$ such that*

$$\|\widehat{v} - \widehat{\mathcal{I}}\widehat{v}\|_{r,\widehat{T}} \leq C|\widehat{v}|_{m,\widehat{T}}, \qquad \forall \widehat{v} \in H^r(\widehat{T}), \quad 0 \leq r \leq m. \tag{2.5}$$

Using the results of Lemma 2.1 and Lemma 2.2, we can derive the following interpolation error estimate:

$$\begin{aligned} |v - \mathcal{I}v|_{1,T} &\leq C \, \|B^{-1}\| \, |\det B|^{1/2} \, |\widehat{v} - \widehat{\mathcal{I}}\widehat{v}|_{1,\widehat{T}} \\ &\leq C \, \|B^{-1}\| \, |\det B|^{1/2} \, |\widehat{v}|_{m,\widehat{T}} \\ &\leq C \, \|B^{-1}\| \, |\det B|^{1/2} \, \|B\|^m \, |\det B|^{-1/2} \, |v|_{m,T} \\ &\leq C \left(\|B^{-1}\| \, \|B\|\right) \, \|B\|^{m-1} \, |v|_{m,T}. \end{aligned} \tag{2.6}$$

The above result highlights the fact that the norms $\|B\|$ and $\|B^{-1}\|$ enter naturally in the error bound on the right and are multiplicatively scaled by the H^m seminorm of v. We note here that the condition number of B, $\kappa(B) = \|B^{-1}\| \, \|B\|$, appears in the right hand side and is related to the geometric measure involving the inscribed and circumscribed circle radii.

2.2 Explicit Residual Error Estimates

In this subsection, we consider the diffusion equation as a model problem to review the methodology to derive error estimates in terms of the residual (see [40]). Let Ω be an open bounded domain and $\bar{t}$ a positive real number. The diffusion equation reads:

$$\partial_t u = \nu \Delta u, \qquad \text{in } \Omega \times (0, \bar{t}), \tag{2.7}$$

where ν defines the viscosity. We shall also prescribe the following boundary and initial conditions:

$$\begin{aligned} &u(\boldsymbol{x}, t) = 0, && \forall \boldsymbol{x} \in \partial\Omega, \quad \forall t \in (0, \bar{t}) \\ &u(\boldsymbol{x}, 0) = u^0(\boldsymbol{x}), && \forall \boldsymbol{x} \in \Omega. \end{aligned} \tag{2.8}$$

The equivalent weak form of the diffusion problem is: Find $u \in L_2(0,\bar{t};V)$ where $V = H_0^1(\Omega)$, such that, for almost all $t \in (0,\bar{t})$,

$$(\partial_t u, v) = -\nu(\nabla u, \nabla v) \qquad \forall v \in V \quad \text{and} \quad u = u^0 \text{ at } t = 0. \tag{2.9}$$

One requirement here is to solve the problem in an explicit manner, using, for example, the forward Euler scheme. In this case, denoting the timestep $\Delta t = t^n - t^{n-1}$, this reduces to: Find $u^n \in V$, $n = 1, 2, \ldots$, such that, given u^0

$$(u^n, v) = (u^{n-1}, v) - \Delta t\ \nu(\nabla u^{n-1}, \nabla v), \qquad \forall v \in V. \tag{2.10}$$

Then, applying the Galerkin approach and considering a conforming finite element subspace V^h of V, we compute a finite element solution $u_h^n \in V^h$ as follows: Given u_h^0, find $u_h^n \in V^h$, $n = 1, 2, \ldots$, such that

$$(u_h^n, v) = (u_h^{n-1}, v) - \Delta t\ \nu(\nabla u_h^{n-1}, \nabla v), \qquad \forall v \in V^h,$$

where u_h^0 is an L^2-projection of u^0 on V^h.

We now investigate the numerical error in u_h^n due to the finite element discretization. In other words, we want to estimate the error $e^n = u^n - u_h^n$. Replacing u^n by $u_h^n + e^n$ in (2.10), the error is shown to be governed by the following equation:

$$(e^n, v) = \mathcal{R}_h(v) + \mathcal{R}_e(v), \qquad \forall v \in V, \tag{2.11}$$

where $\mathcal{R}_h$ and $\mathcal{R}_e$ denote the residual functionals:

$$\begin{aligned} \mathcal{R}_h(v) &= -(u_h^n - u_h^{n-1}, v) - \Delta t\ \nu(\nabla u_h^{n-1}, \nabla v), \\ \mathcal{R}_e(v) &= (e^{n-1}, v) - \Delta t\ \nu(\nabla e^{n-1}, \nabla v). \end{aligned}$$

The residual $\mathcal{R}_h$ is identified with the source of error due to the finite element discretization, while the residual $\mathcal{R}_e$ describes the accumulation of error with time. In this project, we are interested in the effect of $\mathcal{R}_h$ only, so that we look at the component of the error which satisfies:

$$(e^n, v) = \mathcal{R}_h(v). \tag{2.12}$$

We assume here that the component due to the residual $\mathcal{R}_e$ is kept small, using for example an error control strategy during the whole solution procedure.

Following the standard approach used in elliptic problems, we decompose the residual functional into elementwise contributions:

$$\mathcal{R}_h(v) = \sum_{K=1}^{N_e} \Delta t \int_{\Omega_K} r_K v \,\mathrm{d}x + \Delta t \int_{\partial\Omega_K} J_{\gamma,K} v \,\mathrm{d}s \tag{2.13}$$

with r_K and $J_{\gamma,K}$ denoting the interior and inter-element residuals:

$$r_K = -\frac{u_h^n - u_h^{n-1}}{\Delta t} + \nu \Delta u_h^{n-1}$$

$$J_{\gamma,K} = \begin{cases} -\nu\, \mathbf{n} \cdot \nabla u_h^{n-1} & \text{if } \gamma \in \partial\Omega_K \cap \partial\Omega \\ -\frac{\nu}{2}\left[(\mathbf{n} \cdot \nabla u_h^{n-1})_K + (\mathbf{n} \cdot \nabla u_h^{n-1})_L\right] & \text{if } \gamma \in \partial\Omega_K \backslash \partial\Omega \end{cases}$$

where the subscript L stands for the element neighbor Ω_L sharing the edge γ with Ω_K.

Then, observing that the residual $\mathcal{R}_h$ vanishes on V^h, i.e., $\mathcal{R}_h(v) = 0$, $\forall v \in V^h$ (orthogonality relation), and using standard interpolation estimates, we are able to derive the following bounds:

$$\begin{aligned}
(e^n, v) &= \Delta t \sum_{K=1}^{N_e} \int_{\Omega_K} r_K v \,\mathrm{d}x + \int_{\partial\Omega_K} J_{\gamma,K} v \mathrm{d}s \\
&= \Delta t \sum_{K=1}^{N_e} \int_{\Omega_K} r_K (v - \mathcal{I}_h v) \,\mathrm{d}x + \int_{\partial\Omega_K} J_{\gamma,K} (v - \mathcal{I}_h v) \mathrm{d}s \\
&\leq \Delta t \sum_{K=1}^{N_e} \|r_K\|_{0,K} \|v - \mathcal{I}_h v\|_{0,K} + \|J_{\gamma,K}\|_{0,\partial K} \|v - \mathcal{I}_h v\|_{0,\partial K} \\
&\leq C \Delta t \sum_{K=1}^{N_e} \|r_K\|_{0,K} \|v\|_{0,\widetilde{K}} + h_K^{1/2} \|J_{\gamma,K}\|_{0,\partial K} \|v\|_{1,\widetilde{K}} \\
&\leq C \Delta t \sqrt{\textstyle\sum_{K=1}^{N_e} \|r_K\|_{0,K}^2 + h_K \|J_{\gamma,K}\|_{0,\partial K}^2} \times \sqrt{\|v\|_0^2 + \|v\|_1^2}
\end{aligned}$$

where C is a constant independent of h, $\widetilde{\Omega}_K$ a patch of elements sharing one of the nodes of Ω_K, and $\mathcal{I}_h$, the interpolation operator with respect to the finite element space V^h (in the previous subsection, the interpolation operator $\mathcal{I}$ was dedined on a single element). We also point out that, here and elsewhere, we denote norms $\|\cdot\|_{m,\Omega_K}$ as $\|\cdot\|_{m,K}$ for simplicity in notation.

The next step in order to obtain an L_2-norm error estimate would be to replace v by e^n in the relation above, so that:

$$\|e^n\|_0^2 = (e^n, e^n) \leq C \Delta t \sqrt{\textstyle\sum_{K=1}^{N_e} \|r_K\|_{0,K}^2 + h_K \|J_{\gamma,K}\|_{0,\partial K}^2} \times \sqrt{\|e^n\|_0^2 + \|e^n\|_1^2}.$$

Unfortunately, there is no immediate result which proves that $\|e^n\|_1$ is bounded by $\|e^n\|_0$. Nevertheless, one remarks that the time discretization scheme is explicit in time and therefore should be conditionally stable. Actually, in order for the scheme to be stable, every perturbation $\psi \in V$ to a solution of (2.10) should satisfy the stability condition:

$$|\psi|_1^2 < \frac{2}{\Delta t\, \nu} \|\psi\|_0^2. \tag{2.14}$$

If the error e^n is now viewed as a perturbation to the solution, it should then satisfy the above stability condition in the sense that:

$$|e^n|_1^2 < \frac{2}{\Delta t\ \nu}\|e^n\|_0^2. \tag{2.15}$$

Hence, we get

$$\|e^n\|_0^2 + \|e^n\|_1^2 = 2\|e^n\|_0^2 + |e^n|_1^2 \leq 2\|e^n\|_0^2 + \frac{2}{\Delta t\ \nu}\|e^n\|_0^2 \leq 2\left(1 + [\Delta t\ \nu]^{-1}\right) \times \|e^n\|_0^2,$$

so that

$$\|e^n\|_0 \leq C\Delta t\sqrt{2\left(1 + [\Delta t\ \nu]^{-1}\right)}\left\{\sum_{K=1}^{N_e} \|r_K\|_{0,K}^2 + h_K\|J_{\gamma,K}\|_{0,\partial K}^2\right\}^{1/2}. \tag{2.16}$$

This provides us with an explicit error estimate in the L_2-norm. In the case where the space V^h consists of continuous piecewise linear basis functions, the term Δu_h^n in the interior residual r_K naturally vanishes, so the predominant term in the indicator above should be given by $J_{\gamma,K}$. Since the term $J_{\gamma,K}$ is related to the inter-element fluxes, this analysis suggests the use of the edge flux-jump based or patch-recovery indicators described below.

2.3 Edge Flux-Jump Based Indicator

The rationale behind the *flux-jump based indicator* is that the inter-element fluxes are in general discontinuous. Indeed, most finite element simulations use C^0 approximations, for which the finite element solutions are piecewise continuous whereas their derivatives may be discontinuous at the inter-element boundaries. Therefore a major source of the approximation error originates in the jumps in the normal flux or stress at each inter-element boundary and, from the explicit residual error estimate, it seems reasonable to consider a measure of these jumps as an error indicator.

The cost of such an indicator is relatively low as it is straightforward to compute the jump in the stresses at the interface of two elements. Consider an element T and its neighbors T_i, $i = 1, 2, 3$ (see Fig. 2). The jumps in the stresses are given by:

$$[\sigma_i^n] = \sigma_{T,T_i} - \sigma_{T_i,T} \quad , \quad i = 1, 2, 3. \tag{2.17}$$

An obvious choice for the error measure is to take the L_2 norm of the jumps. We obtain an error indicator on each element by equally distributing the quantity above among the two elements sharing the edge, that is:

$$\|[\sigma^n]\|_{\partial\Omega_e} = \left(\int_{\boldsymbol{x}_1}^{\boldsymbol{x}_2}\left(\frac{[\sigma_1^n]}{2}\right)^2 ds + \int_{\boldsymbol{x}_2}^{\boldsymbol{x}_3}\left(\frac{[\sigma_2^n]}{2}\right)^2 ds + \int_{\boldsymbol{x}_3}^{\boldsymbol{x}_1}\left(\frac{[\sigma_3^n]}{2}\right)^2 ds\right)^{1/2}. \tag{2.18}$$

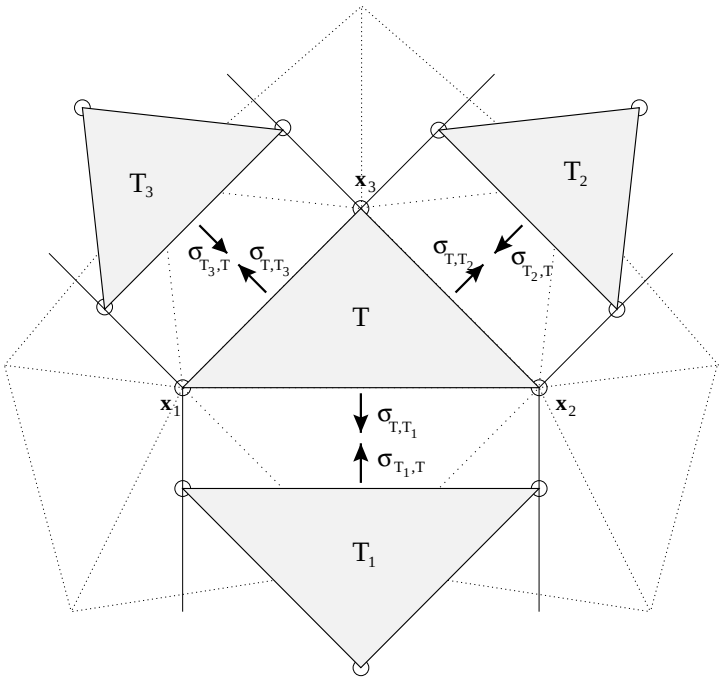

Figure 2: Quantities involved in flux-jump indicators.

2.4 Patch-Recovery Indicator

This section outlines the implementation of a recovery post-processing type of error indicators [48, 51]. Let u_h be a continuous piecewise linear finite element approximation of the solution u of a given boundary-value problem. The objective is to calculate a continuous piecewise linear vector function that will be an approximation of the gradient of u. More, specifically, a recovery operator G_h is defined such that

$$G_h[u_h] \approx \nabla u.$$

Let $\boldsymbol{x}_m$ define one of the vertices of some element in the triangulation $\mathcal{T}_h$. First, we need to identify the patch Π_m consisting of the elements having a vertex at $\boldsymbol{x}_m$; i.e.

$$\Pi_m = \{K \in \mathcal{T}_h : \ \boldsymbol{x}_m \text{ is a vertex of } K\},$$

see Fig. 3. Now, let $\mathbf{z}_K$ denote the barycenter of the element K and $\mathbf{g}$ a vector function with piecewise linear components defined on Π_m; i.e.

$$\mathbf{g}(\boldsymbol{x}) = \begin{pmatrix} a_1 x + b_1 y + c_1 \\ a_2 x + b_2 y + c_2 \end{pmatrix}.$$

The next step is to determine the coefficients a_i, b_i, c_i, $i = 1, 2$, by calculating the discrete least squares fit to the gradient of u_h, sampled at the barycenters of all elements in the patch Π_m. In other words, the coefficients form the unique set of values which minimize the

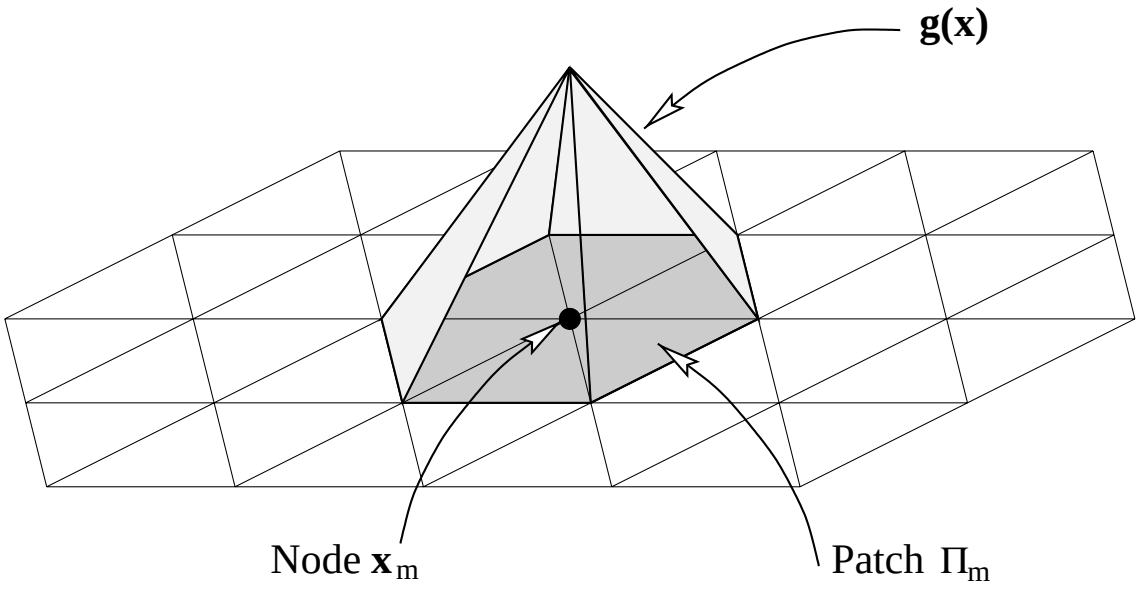

Figure 3: The patch of elements Π_m.

functional:

$$\mathcal{G}(\mathbf{g}) = \sum_{K \in \Pi_m} \left(\mathbf{g}(\mathbf{z}_K) - \nabla u_h(\mathbf{z}_K)\right)^2 .$$

Then the value of $G_h[u_h]$ at the node $\boldsymbol{x}_m$ is set to be

$$G_h[u_h](\boldsymbol{x}_m) = \mathbf{g}(\boldsymbol{x}_m).$$

Applying the above procedure to all nodes in the triangulation provides the recovery operator $G_h[u_h]$. In the case where the node $\boldsymbol{x}_m$ lies on the boundary of the domain, we may not have enough neighbor elements; i.e., the discrete least squares fit may not provide a unique solution. If so, we enlarge the patch by adding all elements that are vertex-neighbors to the neighbors.

To obtain an error indicator, the recovery operator is employed as above to compute a recovered stress. Then, the difference between the recovered stress $\widehat{\sigma}$ and the stress σ_h obtained from the finite element approximation may be used to estimate the error. In the later computations we use the von Mises stress. That is, we compute $(\widehat{\sigma} - \sigma_h)_{vm}$ on each element, where the von Mises stress is defined as

$$\sigma_{vm} = \sqrt{0.5\left[(\sigma_x - \sigma_y)^2 + (\sigma_x - \sigma_z)^2 + (\sigma_y - \sigma_z)^2\right] + 3(\tau_{xy}^2 + \tau_{xz}^2 + \tau_{yz}^2)}.$$

The element indicator is then given by $(\widehat{\sigma} - \sigma_h)_{vm}$ measured in the L_2-norm. Alternatively, we could use the recovered stress $\widehat{\sigma}$ and strain $\widehat{\epsilon}$ to compute an *energy* type error indicator

$$\eta_K^2 = \int_K (\widehat{\sigma} - \sigma_h)(\widehat{\epsilon} - \epsilon_h)\, dx,$$

and stipulate that $\eta_K \approx \|e\|_K$.

3 Element Deformation and Shape Control

Element shape distortion, due to either mesh motion or mesh adaption, often degrades the accuracy in numerical approximations, as well as the efficiency and reliability of simulations. For example, it is well known that triangles or tetrahedra with large obtuse angles have inferior approximation properties. Slender elements and sudden mesh size gradations also lead to ill-conditioning problems and severe restrictions on the timestep. This latter issue is particularly relevant in the context of explicit Lagrangian impact problems. *Element shape quality* is thus becoming a research topic of growing importance. Recently, several choices of *shape quality metrics* have been proposed which aim at quantifying the element distortion. These include maximum and minimum angles, ratio of incircle to circumcircle diameters, ratios involving edge lengths and element areas in two dimensions. We refer the reader to [16, 25] for more details.

In standard finite element implementations, elements in the physical domain are images of a *reference element* via mappings. The most frequently used mappings are the parametric mappings which employ the same element basis functions as in the finite element approximation, with the nodes regularly spaced on the reference element. If the map is not affine, when dealing, for instance, with quadrilateral elements or curvilinear elements, the map is not necessarily bijective. This is observed, for instance, when elements in the physical domain have re-entrant corners. On the other hand, the mappings in the case of triangles with straight edges are always invertible provided the elements are not degenerate (the node points cannot be aligned). In this case, one can use the Jacobian of the transformation and of its inverse to characterize the distortion resulting from the maps to and from the reference element. We elaborate on this point below and extend the idea to the analysis of transient deforming mesh problems. We remark here that the Jacobian matrix, its determinant as well as particular matrix norms, were actually used in [25] to study the effect of distortion and mesh quality issues.

Although the issue of element shape quality is present in mesh adaption, for example when the nodes of the mesh are redistributed in order to equidistribute the numerical error, this issue is even more acute for Lagrangian simulations. In this instance, the grid points are allowed to move, while retaining the connectivity, and one has to make sure that the triangles do not overlap one another or do not get folded with reversed node numbering due to edge tangling. Even if there is no tangling, we still need to control the time evolution of the elements to avoid ill-shaped or degenerate elements. To date, several metrics have been proposed in the literature and have been the subject of recent studies [15, 45, 50]. In particular, the following were studied in [16] for an investigation on grid degeneracy arising in adaptive refinement of tetrahedra:

$$f_W = \frac{\textit{minimum distance from the centroid of the element to the faces}}{\textit{length of the longest edge}} \tag{3.19}$$

$$f_S = \frac{\textit{volume of the element}}{(\textit{length of the longest edge})^d} \tag{3.20}$$

$$\rho = \frac{\textit{radius of the inscribed sphere}}{\textit{radius of the circumsphere}} \tag{3.21}$$

where d is the space dimension.

More recently, mesh quality indicators have been examined in terms of objective functions for attaining valid, high quality meshes in the framework of three-dimensional mesh optimization [25]. The approach uses element Jacobian and matrix norms to build suitable three-dimensional objective functions. It is also shown that some relations between objective functions in dimension two do not hold in dimension three. Although the most logical choice is to define non-dimensional objective functions, those have barriers in the sense that they become infinite for a certain set of matrices. One attractive choice is to use the element condition number:

$$\kappa(B) = \|B\|_F \, \|B^{-1}\|_F \ , \tag{3.22}$$

where B, as in Section 2.1 represents the Jacobian of the mapping from the reference element $\widehat{T}$ to the element in the physical domain T. In (3.22), the norm $\|\cdot\|_F$ is called the Frobenius matrix norm and is defined as:

$$\|B\|_F = \left[tr(B^T B)\right]^{1/2} \ .$$

It is argued that minimizing the condition number of B (within the framework of mesh optimization) seems an appropriate choice as it measures the distance of B from the set of singular matrices. Interestingly, we observe that a term similar to the condition number of B appears in the derivation of the *interpolation error estimates* (2.6), which justifies the use of the element condition number as a mesh quality measure.

Most previous work using shape indicators has focussed on the static grid problem. In this case, unstructured meshes with poorly shaped elements may be generated through adaptive refinement based on error estimates. The mesh can be improved using edge/face swaps in a Delaunay scheme or smoothing with node redistribution.

In the case of the Lagrangian impact analysis, element distortion is also caused by the evolution of the solution to follow specific features, by the change in the domain and its boundary, and by displacement of the nodes resulting from the Lagrangian formulation. In such situations, one may also be interested in the rate at which the local quality of any element is deteriorating (essentially a *look ahead* issue). A similar issue can be raised with respect to repeated local refinements. This suggests that one may want to control both the mesh at a given time as well as the rate of change of the local cell quality after one timestep. This latter point may be crucial when the nodes move rapidly within one timestep. This

idea is explored later in the numerical studies of hybrid indicators by relating the *relative velocity* of the nodes to the rate of change of the local mapping. Accordingly, in the present work, we restrict our attention to shape quality indicators based on the map and its local rate of change.

Remark *The acceptability of an element with a given shape seems to be problem dependent as well as perhaps method dependent. This is certainly true when "ill-shaped" elements are in parts of the domain where the behavior of the solution is rather benign. Examples of such grids are actually common in the "moving grid" literature [32]. Therefore, in the context of numerical approximation, the concept of shape quality indicator is itself flawed. Moreover, an error indicator computed using local approximation of the solution should implicitly encompass the effect of element shape on the error. This is more than an issue of semantics and it serves to emphasize that the error in this context can not be separated into its contributing sources so our constructions of hybrid indicators should be viewed in this pessimistic light.*

On the other hand, it is not unreasonable to argue that the occurrence of a large error or large value of an error indicator on a poorly shaped cell bears further scrutiny. Maybe the local element quality indicator and the local error indicator should be examined separately. This is part of the motivation for the additive hybrid indicator considered later. The multiplicative indicator is suggested formally from the structure of the interpolation error estimate and the entry of the forward and reverse maps in the coefficient of the estimate. Finally, the effect of the shape and size of the element may be more important in relation to solver efficiency and ill-shaped elements detected and corrected for this reason alone.

3.1 Growth of Shape Error

We consider here another issue of interest, namely the motion of nodes which often result in mesh tangling. It may be desirable to avoid tangling either by considering a penalty approach, or, by underrelaxing the mesh motion selecting only a fraction of the predicted deformation step. Estimating the local rate of deformation would help to ascertain which elements are deforming rapidly and take appropriate action. One such indicator can be obtained by computing the rate of change of the Jacobian matrix with respect to time.

Using the simple geometric shape quality measure ρ given in (3.21), we show how to calculate its rate of change, denoted subsequently by $\dot{\rho}$, for any element in the mesh. We expect this measure to inform us about the elements which are rapidly deteriorating. Moreover, integrating through a timestep, we have the approximation at the end of the step for the shape quality indicator.

When the elements change their shape with time, as in Lagrangian simulations, the mappings F become a function of time, that is $F = F(t)$. Therefore, the variation of $\|B\|$ and $\|B^{-1}\|$ indicates how the element shape changes and should provide a relevant indicator of element deterioration. We believe that an error indicator for Lagrangian simulations should include measures of $\|B\|$ and $\|B^{-1}\|$ as well as their growth. We now develop such an indicator.

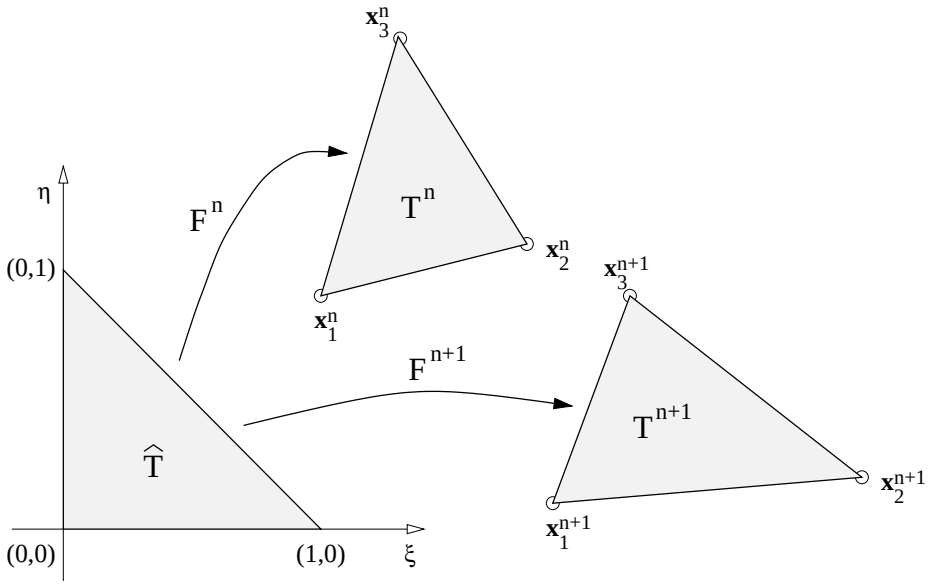

Figure 4: Successive maps defining a deforming finite element at times t^n and t^{n+1}.

Let T^n and T^{n+1} represent two configurations of an element at the successive times t^n and t^{n+1} as shown in Fig. 4. At any time, the entries of the matrix $B(t)$ are given by $b_{ij}(t)$ as in (2.2). From the Taylor series representation, we know that there exists a time $\tau \in [t^n, t^{n+1}]$ such that:

$$b_{i,j}(t^{n+1}) = b_{i,j}(t^n) + \Delta t\, \dot{b}_{i,j}(\tau),$$

where Δt denotes the timestep, i.e. $\Delta t = t^{n+1} - t^n$. Then, using any matrix norm, we have:

$$\|B^{n+1}\| \leq \|B^n\| + \Delta t \max_{t\in[t^n,t^{n+1}]} \left(|\dot{x}_2 - \dot{x}_1|, |\dot{x}_3 - \dot{x}_1|, |\dot{y}_2 - \dot{y}_1|, |\dot{y}_3 - \dot{y}_1|\right) \ . \tag{3.23}$$

The interpolation error estimate (2.6) together with (3.23) leads to the following result:

Theorem 3.1 *Let T^n and T^{n+1} denote a same element at times t^n and t^{n+1} defined by the affine maps F^n and F^{n+1} from a reference element $\widehat{T}$ given by (2.1) with corresponding matrices B^n and B^{n+1}. Let $v \in H^r(T^{n+1})$ and $\mathcal{I}$ denote the finite element interpolation operator as defined in Section 2.1. Then the interpolation error $v - \mathcal{I}v$ satisfies*

$$\begin{aligned} |v - \mathcal{I}v|_{1,T^{n+1}} \leq\ & C\ \|(B^{n+1})^{-1}\|\ \|B^{n+1}\| \\ & \times \left(\|B^n\| + \Delta t \max_{t\in[t^n,t^{n+1}]} \left(|\dot{x}_2 - \dot{x}_1|, |\dot{x}_3 - \dot{x}_1|, |\dot{y}_2 - \dot{y}_1|, |\dot{y}_3 - \dot{y}_1|\right) \right) \\ & \times \|v\|_{2,T^{n+1}} \end{aligned} \tag{3.24}$$

where C is independent of the shape of the element. The quantities $\dot{x}_i$, $\dot{y}_i$, $i = 1, 2, 3$, in (3.24) denote the x and y velocity components of the element vertices.

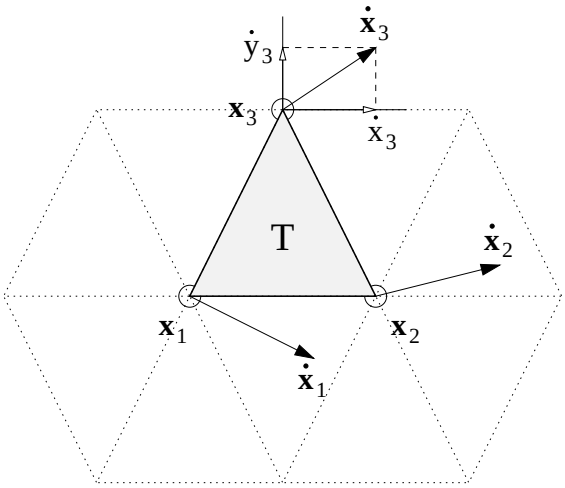

Figure 5: Triangular element with nodal velocity vectors.

The first factor on the right in (3.24) is precisely the quality measure advocated in [25] that is based on the element condition number shape matrix. The second factor involves the relative velocities in (3.23) and can be interpreted as an indicator of the rate of change of the element shape.

We now explore several possibilities for determining error and shape quality indicators. These are (1) an indicator based on relative velocities, (2) an additive hybrid indicator and (3) a multiplicative hybrid indicator.

3.2 $\mathbf{RV_S}$ Indicator

The first refinement/error indicator we implement is based on characterizing the interpolation error due to mesh distortion as outlined above. It may alternatively be motivated from simple physical reasoning. Large and rapid mesh deformation is the primary difficulty when dealing with Lagrangian impact simulations and can cause computation breakdowns or simply large errors. Therefore an effective precursor to such deformations will be an indicator for guiding mesh refinement/unrefinement. From the form of the bounds in (3.23) and (3.24), the relative velocity of the adjacent nodes can be used as an indicator of change in shape quality. Rather than the maximum on $[t^n, t^{n+1}]$, one may consider other norms at either t^n or t^{n+1} in practice. We consider below the Euclidean norm of the relative velocities based on the relative velocities of the nodes on the respective sides of the element. Then, computing the sum of each nodal indicators for each element will provide a simple predictor of distortion error for large mesh deformation. For the typical element T shown in Fig. 5, with associated nodes N_1, N_2 and N_3, we then define

$$\begin{aligned} \mathrm{RV}_{12} &= \sqrt{(\dot{x}_2 - \dot{x}_1)^2 + (\dot{y}_2 - \dot{y}_1)^2} \\ \mathrm{RV}_{23} &= \sqrt{(\dot{x}_3 - \dot{x}_2)^2 + (\dot{y}_3 - \dot{y}_2)^2} \\ \mathrm{RV}_{31} &= \sqrt{(\dot{x}_1 - \dot{x}_3)^2 + (\dot{y}_1 - \dot{y}_3)^2} \end{aligned} \tag{3.25}$$

and

$$\mathrm{RV}_{\mathrm{S}} = \mathrm{RV}_{12} + \mathrm{RV}_{23} + \mathrm{RV}_{31}. \tag{3.26}$$

The quantity RV_{S} is clearly a *look ahead* indicator for evolving large mesh deformations. This indicator is actually similar in spirit to the indicators proposed in [8, 14].

3.3 Additive and Multiplicative Hybrid Indicators

One goal in the present work is to control the approximation error and the element shape quality simultaneously. This can, of course, be done sequentially in a simulation step – checking the error indicators to guide mesh refinement and coarsening and then the element shape indicators to guide node smoothing and local modification of the mesh. This may indeed be the preferred approach. However, it is also possible to construct various *hybrid* indicators that combine the above indicators into a single metric. Since the strategies for improving the grid in these respective situations are different it is debatable whether such a hybrid approach has merit. Our work here on this issue should be regarded as exploratory in this sense.

Conceptually, the idea may be thought of in a similar way to grid optimization strategies which use an objective function that has several weighted mesh attributes [12, 15, 39]. Likewise, in hybrid indicators, we attempt to combine the benefits of the two previous types of indicators using appropriate scaled and weighted combinations. We consider both an additive and a multiplicative hybrid indicator. In some respects the latter is more natural from the form of the interpolation estimate described earlier in (3.24), where the multiplicative coefficients depend on the maps that are evolving in time and characterize the deformation. On the other hand, the additive form is closer to the weighted average in the optimization problems mentioned above.

Additive Combination

In the additive combination, we construct the refinement indicator for the element Ω_e using a linear combination of the two previously defined relative velocities and flux-jump indicators:

$$H_e = \alpha \, \frac{(\mathrm{RV}_{\mathrm{S}})_e}{(\mathrm{RV}_{\mathrm{S}})_{\max}} + \beta \, \frac{\|[\sigma^n]\|_{\partial\Omega_e}}{\|[\sigma^n]\|_{\partial\Omega_e \max}} \tag{3.27}$$

where α, β are empirically chosen scalars and the maximum values over the entire grid are used for normalization.

Multiplicative Combination

In the multiplicative combination, the refinement indicator is obtained as the product of the $\mathrm{RV_S}$ and flux-jump indicators at the element level. We can write this indicator as

$$E_e = \frac{(\mathrm{RV_S})_e}{(\mathrm{RV_S})_{\max}} \times \frac{\|[\sigma^n]\|_{\partial\Omega_e}}{\|[\sigma^n]\|_{\partial\Omega_e \max}} \tag{3.28}$$

Both hybrid indicators are tested in the Lagrangian impact simulations.

4 Numerical Studies

4.1 Application and Formulation

Consider a body occupying a reference configuration Ω^n at time t^n and loaded incrementally over the timestep Δt to change the deformation d^n to $d^{n+1} = d^n + u$ at $t^{n+1} = t^n + \Delta t$. Here $u \in V$, the space of kinematically admissible displacements. The incremental variational form of the equilibrium equation obtained using the virtual work principle is: Find $u \in V$ satisfying the essential boundary conditions and initial conditions such that

$$\int_{\Omega^n} \sigma^{n+1}(u) : \nabla_0 v \, \mathrm{d}x - \int_{\Omega^n} (f^{n+1} - \rho^n a^{n+1}) \cdot v \, \mathrm{d}x - \int_{\partial\Omega^n} g^{n+1} \cdot v \, \mathrm{d}s = 0, \qquad \forall v \in V, \tag{4.29}$$

where σ^{n+1} is the first Piola-Kirchoff stress tensor at time t^{n+1} and f^{n+1}, a^{n+1}, g^{n+1} are the corresponding body force, acceleration and traction; ρ^n is the mass density in the reference configuration, $v \in V$ is the admissible virtual displacement and ∇_0 is the material gradient. A finite element discretization of (4.29) leads to the approximate problem: Find $u_h \in V_h$ satisfying the prescribed boundary and initial conditions such that

$$\int_{\Omega^n} \sigma^{n+1}(u_h) : \nabla_0 v_h \, \mathrm{d}x - \int_{\Omega^n} (f^{n+1} - \rho^n a^{n+1}) \cdot v_h \, \mathrm{d}x - \int_{\partial\Omega^n} g^{n+1} \cdot v_h \, \mathrm{d}s = 0, \qquad \forall v_h \in V_h, \tag{4.30}$$

with V_h a conforming finite element of V.

Denoting the error $e = u - u_h$, $e \in V$, and using integration by parts on each element, it follows from (4.29) and (4.30) that for all $v \in V$

$$\begin{aligned} \int_{\Omega^n} \sigma^{n+1}(e) : \nabla_0 v \, \mathrm{d}x &= \int_{\Omega^n} (f^{n+1} - \rho^n a^{n+1}) \cdot v \, \mathrm{d}x \\ &\quad - \int_{\Omega^n} \sigma^{n+1}(u_h) : \nabla_0 v \, \mathrm{d}x + \int_{\partial\Omega^n} g^{n+1} \cdot v \, \mathrm{d}s \\ &= \sum_{K \in \mathcal{P}} \int_K r_K(u_h) \cdot v \, \mathrm{d}x + \sum_{\gamma \in \mathcal{E}} \int_\gamma J(u_h) \cdot v \, \mathrm{d}s \end{aligned}$$

where r_K denotes the element interior residual in terms of u_h and J_γ is the jump in the normal stresses $J_\gamma(u_h) = [\sigma] \cdot n = [\sigma^n]$. Here, the set $\mathcal{E}$ denotes the union of all element interfaces.

The stresses $\sigma^{n+1}(u)$ are computed from the primary variable u by considering its hydrostatic and deviatoric components. The hydrostatic part is usually obtained using an equation of state and the deviatoric part from a constitutive law. For example, typical equations of state and constitutive models for metals at low temperatures are:

$$p - p_{\text{ref}} = \rho\,\Gamma\,(e_i - e_{i\text{ref}}) \tag{4.31}$$

and

$$\widehat{\dot{S}_{ij}} = 2\,\mu\,\dot{e}_{ij} \tag{4.32}$$

$$S_{ij} = \begin{cases} \widehat{S_{ij}} & \text{if} \quad \widehat{S_{kl}}\widehat{S_{kl}} < 2Y^2/3 \\ \sqrt{\dfrac{2}{3}}Y\dfrac{\widehat{S_{ij}}}{\widehat{S_{kl}}\widehat{S_{kl}}} & \text{if} \quad \widehat{S_{kl}}\widehat{S_{kl}} \geq 2Y^2/3 \end{cases} \tag{4.33}$$

where (4.31) is the Mie-Gruniesen equation of state and (4.33) is the so-called *radial return* formulation for elastic-plastic deformation. Here, p is the pressure, e_i the internal energy, Γ the Gruniesen coefficient, S and e the deviatoric stress and strain, μ the shear modulus and Y the yield stress.

Equation (4.30) can be rewritten in matrix form as

$$Ma^{n+1} + F_{\text{int}}^{n+1} = F_{\text{ext}}^{n+1} \tag{4.34}$$

where F_{ext}^{n+1} is due to applied body forces and tractions and F_{int}^{n+1} arises from the internal stresses.

The equations of motion are integrated explicitly using the lumped mass matrix approach. Therefore, given the initial velocities and displacements, the velocities and displacements at the next time step are computed using a simple vector-vector multiplication. At each time step, only the element strains and stresses, nodal forces and accelerations need to be integrated to obtain F_{ext}^{n+1} and F_{int}^{n+1}. Complex non-linear constitutive models and equations of state are easily incorporated in this cycle. Moving slide lines are used to account for contacting surfaces. Elements are judged to have failed based on an appropriate physical criteria such as equivalent plastic strain. Failed elements are removed from the simulation in a process called erosion. This is the basic solution process for most explicit dynamics codes, such as EPIC described below.

4.2 Lagrangian simulations and Results with EPIC

The EPIC code, developed by Johnson *et al.* [23, 24], uses an explicit Lagrangian finite element formulation. The code has been used and continually upgraded over the last two

decades. It is by now well tested and its capabilities and limitations well known. Access to a research version of the source code, its simple data structure and robust computational algorithm make it an attractive choice for our preliminary studies.

The first challenge of solution based adaptivity using finite element codes is the design of a data structure capable of supporting the creation and deletion of new elements. The EPIC code and most Lagrangian codes designed for simulation of high velocity impact include the capabilities of element erosion and re-zoning. These features form the basis for designing the extensions necessary to perform automatic element creation and deletion. In particular, the code allows us to exclude elements from a simulation, to delete elements that are marked for refinement, and to add new elements and nodes at the bottom of the list of existing elements and nodes. While this is sufficient for the present work, support of efficient solution adaptive codes will require a dynamic data structure with the ability to allocate and deallocate memory as needed. Furthermore, the tree structure, implicit in many adaptive refinement codes, will have to be maintained to implement coarsening by unrefinement. This capability is essential for wave propagation type problems such as those of interest in these applications.

Adaptivity and error estimation require frequent reference to information from neighboring elements. Indeed, neighbors often need to be refined in order to produce conforming meshes, while information from the neighbors is also required when computing error indicators such as the flux-jump based or patch recovery indicators. The EPIC code was augmented to provide this functionality. Additional data structures were created to store lists of elements associated with each node. The neighbors of an element are efficiently retrieved from this information only.

Since the EPIC code lacks the ability to support hanging nodes, only conforming meshes are allowed. In consequence, neighbors of an element targeted for refinement often need to be refined as well, and so on in a recursive fashion. We will consider three types of refinement on the triangles used to construct the meshes in our test problems. These are shown in Fig. 6 where shaded triangles represent the elements which need to be refined. Type I refinement is used when the element to be refined and its neighbor both share their longest edge, so that it is a simple matter to refine them both. In the case where the element and its neighbor do not share the longest edge, we allow the refinement to propagate through the mesh. We split the longest edge of the neighbor and enforce a split on its neighbor. We then split the neighbor in three and original element in two. We designate this refinement Type II. This restriction to only two levels is indeed somewhat ad-hoc but keeps the program complexity manageable and alleviate the lack of data-structure support. In the third type of refinement, called Type III, an element is split in three by joining its centroid to each node. However, the child elements get worse aspect ratios than the parent element with this type of refinement.

Once an element is refined, the nodal and elementwise values of various solution quantities are interpolated through simple schemes on the new nodes and elements. Interpolation routines are part of the re-zoning capability of the EPIC code. Since new elements are created by splitting old elements, they inherit all element level state variables from the parent element. Nodes derive their state variables from parent elements or nodes.

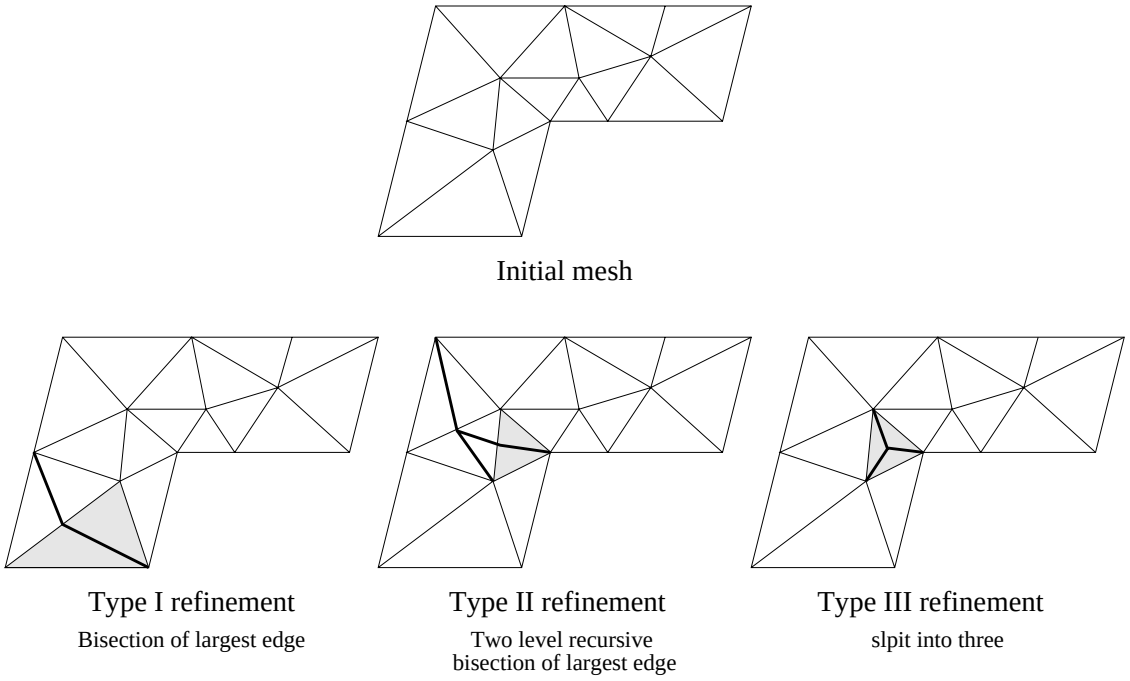

Figure 6: Types of refinement allowed.

Unlike refinement, removing elements to improve the geometry of the remaining elements is rather a complex operation. The classical way of removing elements in a zone and replacing them with a new local remesh called rezoning has been a part of EPIC and similar Lagrangian codes. We experiment here with a slightly different approach. We only allow a few types of local unrefinement where only elements and immediate neighbors are affected (see Fig. 7). Unrefinement is triggered by the same element based measures used for refinement. A more systematic unrefinement scheme where the size of patch to unrefine is determined adaptively will be undertaken in future work.

Finally, the adaptive strategy to be utilized consists of a simple fixed fraction refinement approach, i.e., the elements with refinement indicators in the top 20 or 30 percent are refined after each 25 percent of total simulation time. A better approach would be to trigger refinement when the error indicator exceeds preset tolerances. We are developing better targeted refinement strategies [35] where the error is spatially equidistributed and refinement is monitored to obtain a desired error level.

Numerical results

We have implemented the above refinement indicators and adaptive strategies for the simulation of the classical Taylor-Anvil impact quality test for a cylinder of iron striking a rigid surface (Fig. 8). The initial velocity of the cylinder is set at 152.4 m/s. We show a progressive sequence of results over a time period of 10^{-5} seconds. The initial mesh and geometry are shown in Fig. 8. In this first test the RV_S indicator and refinement strategy are applied. Fig. 9 shows the deformed geometry, the values of the refinement indicator, the plastic strain distribution, and the corresponding adapted mesh at 10^{-5} seconds. This

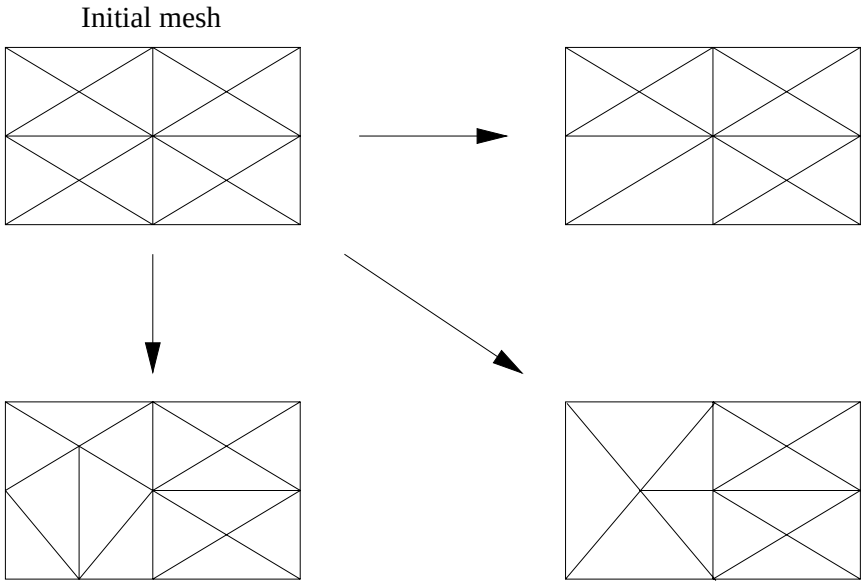

Figure 7: Types of coarsening allowed.

strategy adds 32 nodes and 155 new elements and we observe that the mesh is essentially refined in the region where maximum element distortion is experienced. However, the plots of plastic strain indicate that the highly strained region in the bottom outer zone of the cylinder does not see much refinement.

In the next test, we use the flux jump indicator and the fixed fraction refinement strategy with top 30 percent of the elements being refined. The computations are again carried out over the same time period of 10^{-5} seconds. The results of this simulation are shown in Fig. 10, i.e., the adapted meshes, the contours of $\|[\sigma^n]\|_{\partial\Omega_e}$ and the plastic strain distribution obtained at 10^{-5} seconds.

In the third set of tests, we use the additive hybrid indicator H_e with $\alpha = \beta = 1$ and refinement triggered whenever $H_e > 1$. In Fig. 11, we show the adapted meshes, contours of $\|[\sigma^n]\|_{\partial\Omega_e}$ and $\mathrm{RV_S}$ at 10^{-5} seconds. This indicator and strategy add 55 nodes and 201 elements. We observe that the refinements are now distributed more evenly.

In the fourth set of tests, we use the hybrid indicator E_e with refinement triggered whenever $E_e > 0.3$. This indicator and strategy add 55 nodes and 201 elements. The refinements are now distributed more evenly.

In the next set of results, we investigate the performance of the patch-recovery error indicator described in Section 2.4. Fig. 13 shows the error indicators obtained and refinement at the time 2.25×10^{-5} and 3.0×10^{-5} seconds. Refinement patterns are similar to the other indicators.

Finally, we combine refinement and unrefinement and look at its effect on the element aspect ratios. The objective is to examine how the mesh quality would improve. In Fig. 14, we see that the average aspect ratio is now dramatically improved, but not the minimum aspect ratio. This is due to the refinement/unrefinement strategy rather than the error

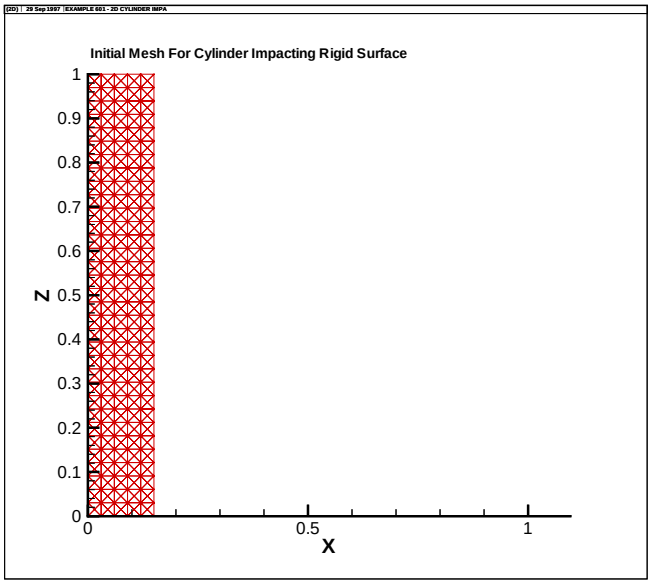

Figure 8: Initial mesh for cylindrical rod impacting rigid surface

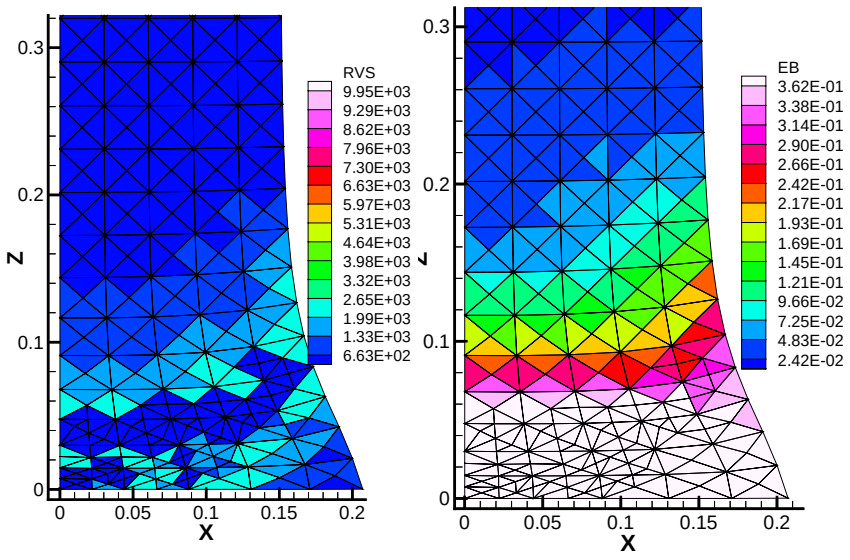

Figure 9: Mesh and contours of RV_S and plastic strain for a cylindrical iron rod impacting rigid surface at 10^{-5} seconds after multiple adaption cycles using the RV_S indicator. Rod initial velocity is 152.4 m/s.

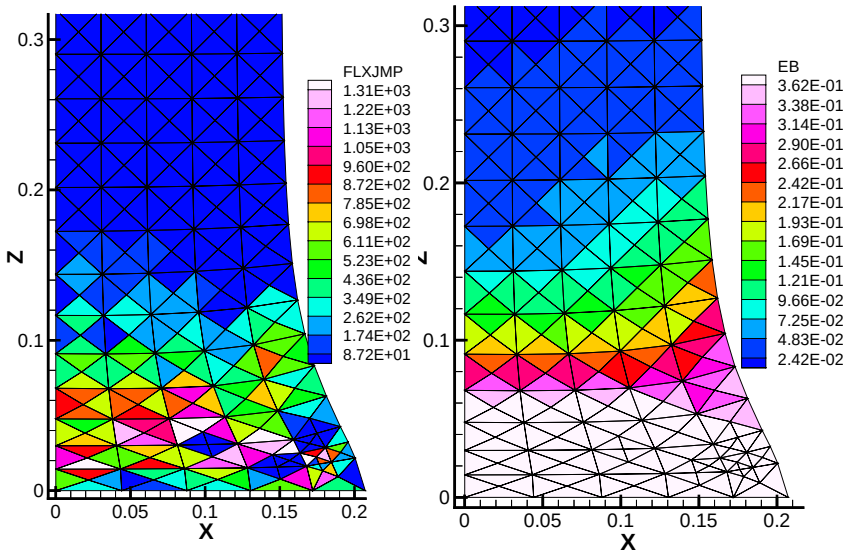

Figure 10: Mesh and contours of flux jump and plastic strain for a cylindrical iron rod impacting rigid surface at 10^{-5} seconds after multiple adaption cycles using the flux jump indicator. Rod initial velocity is 152.4 m/s.

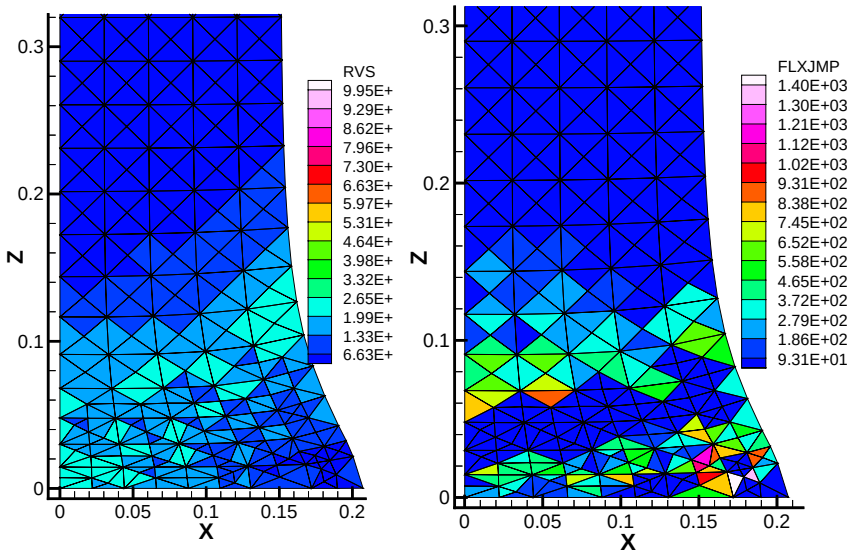

Figure 11: Mesh and contours of $\mathrm{RV_S}$ and flux jump at 10^{-5} seconds for a cylindrical iron rod impacting rigid surface after multiple adaption cycles using the additive hybrid indicator. Rod initial velocity is 152.4m/s.

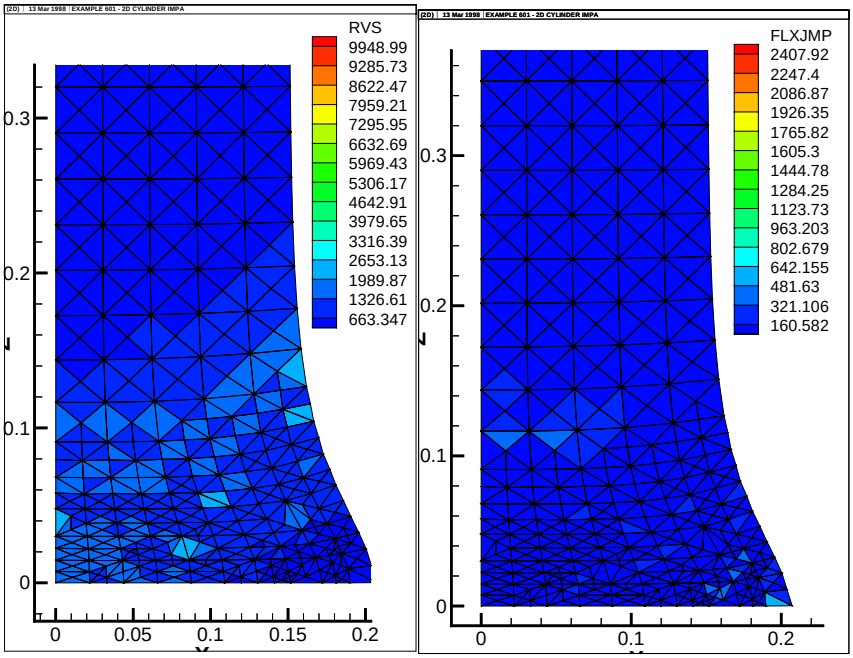

Figure 12: Mesh and contours of RV_S and flux jump at 10^{-5} seconds for a cylindrical iron rod impacting rigid surface after multiple adaption cycles using the multiplicative hybrid indicator. Rod initial velocity is 152.4m/s.

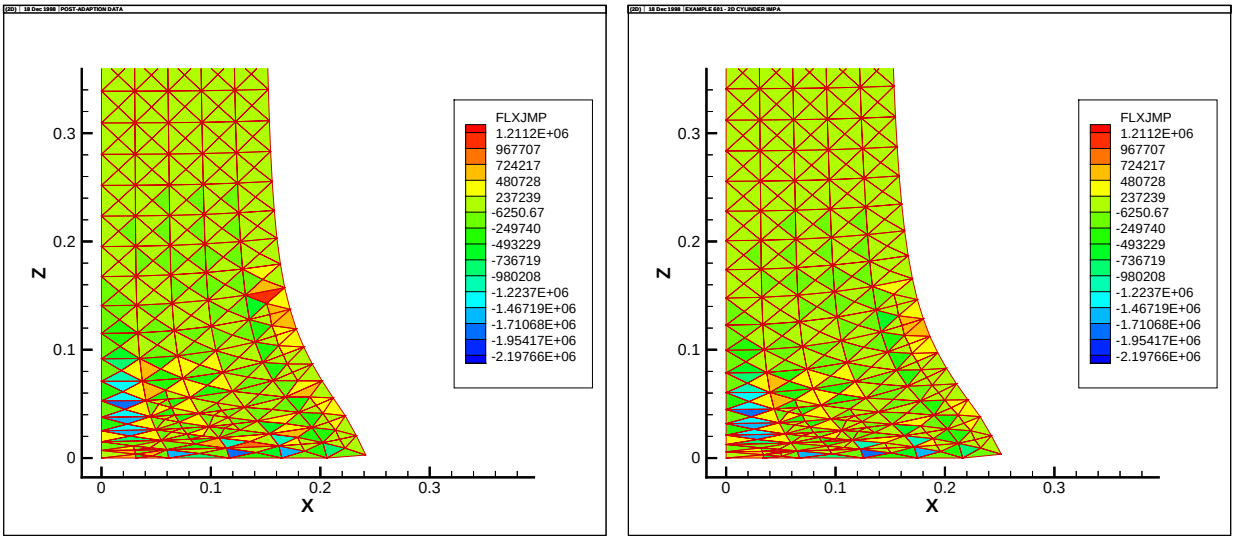

Figure 13: Contours of adapted mesh and energy associated with ZZ type error indicator at 2.25×10^{-5} and 3.0×10^{-5} seconds. Rod initial velocity is 203.2 m/s.

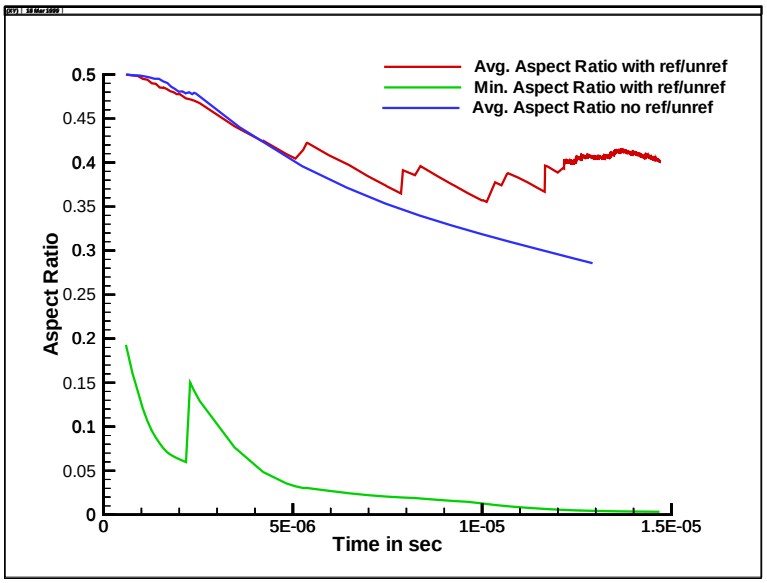

Figure 14: Aspect ratios as the mesh is refined and unrefined.

indicators, since in each case the element with the minimum aspect ratio was identified as a target but not modified due to the refinement constraints.

4.3 Eulerian Simulations and Results with CTH

As a further test of applicability of adaptive mesh refinement to penetration and impact simulations, we have implemented the capability for basic adaptivity into the Eulerian impact physics code CTH. The adaptive strategy is a block-scheme, with the mesh resolution for adjacent blocks strictly limited to 2:1 ratios. A considerable portion of this effort has been involved in the development of supporting algorithms for the treatment of multi-material elements, including block adaptive advection [30] and multi-material refinement [31], as well as the parallel implementation and block-to-block communication [17]. As such, implementation of realistic error indicators, or a test of a variety of indicators, as was shown with the Lagrangian results, has not yet been performed. Nevertheless, preliminary calculations, which have been carried out as part of the code development and verification testing process, have yielded impressive results.

Examples from one of these simulations are shown in Fig. 15, where the propagation of a two-dimensional blast wave through air is shown. The simulation took 1.8 CPU hours to complete. A corresponding uniform mesh version of this problem was also run; it produced almost identical results to the ones shown here but required 145 CPU hours to complete.

A second example is shown in Fig. 16, where the impact of an aluminum sphere at 5 km/s on an aluminum plate is shown. Refinement and unrefinement is seen so that the highest resolution mesh follows the fragments.

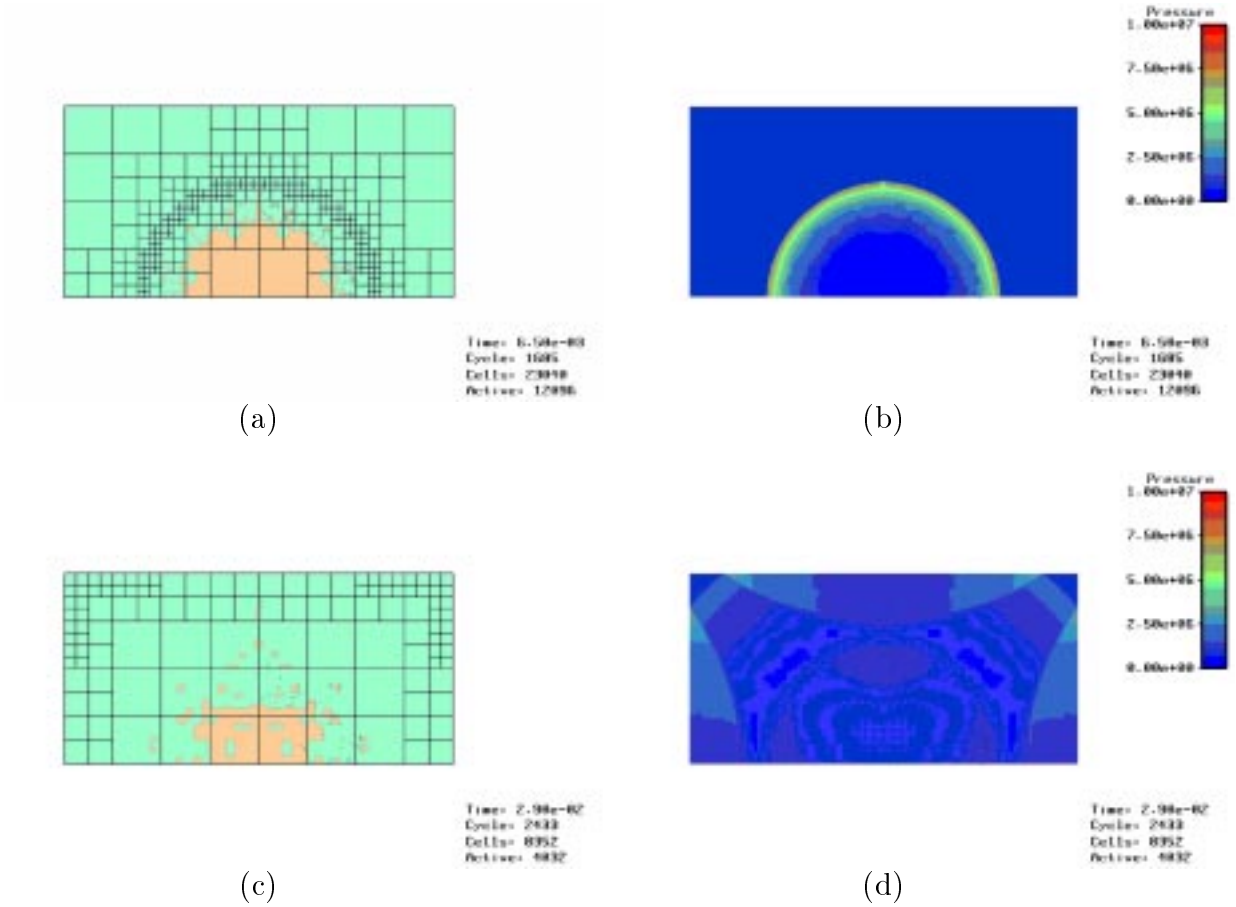

Figure 15: Sample calculation of a two-dimensional blast wave propagating through air and reflecting off rigid walls, showing (a) mesh blocks at 6.5 ms, (b) pressure at 6.5 ms, (c) mesh blocks at 29 ms and (d) pressure at 29 ms. This calculation was run in 1.8 CPU hours, a comparable uniform mesh version of this simulation required 145 CPU hours.

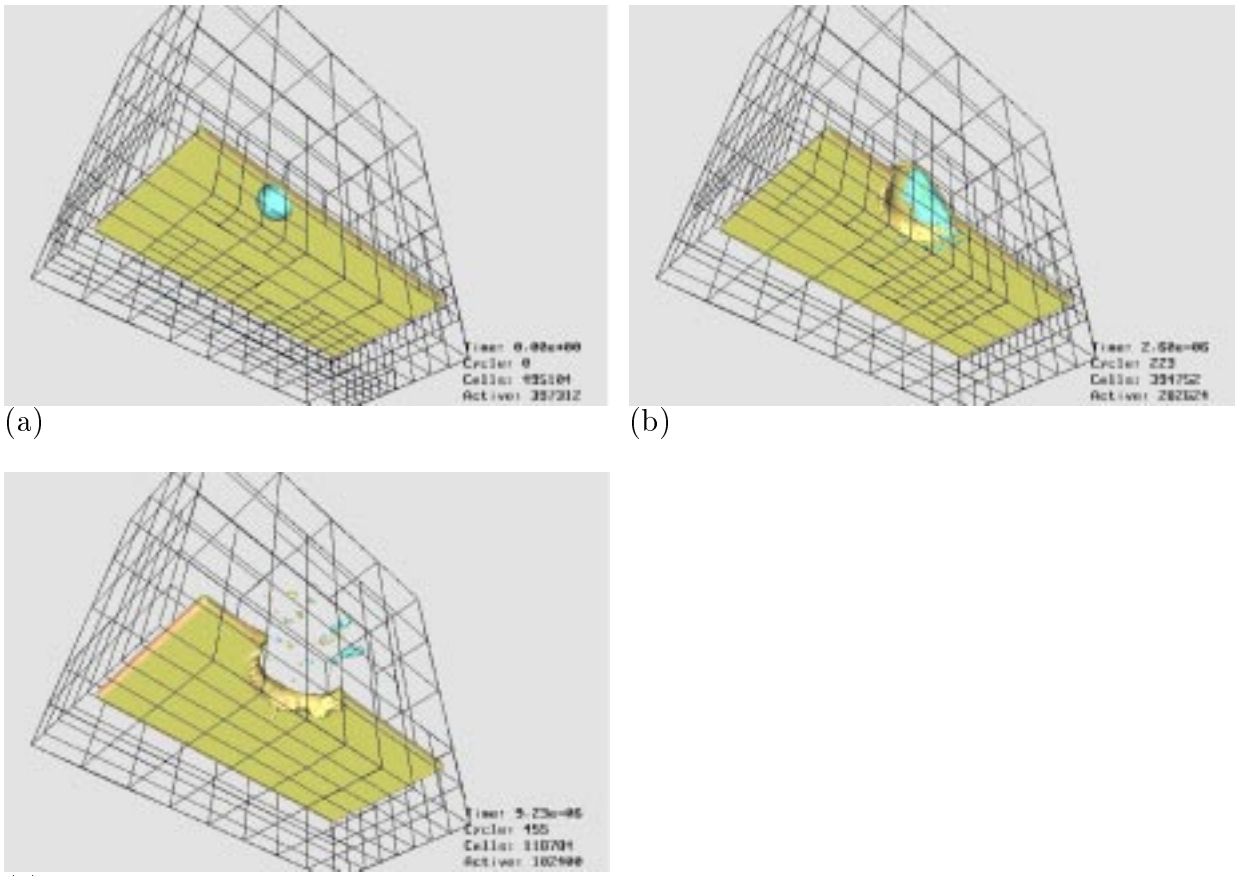

Figure 16: Simulation of the impact of an aluminum sphere onto an aluminum plate at 5 km/s, showing (a) initial configuration of materials and mesh blocks, (b) materials and mesh blocks at $2.6\mu s$ and (c) materials and mesh blocks at $9.2\mu s$.

5 Concluding Remarks

In this work we have described studies on the problem of simultaneously controlling the error in the approximate solution (by adaptive refinement or coarsening) and controlling the shape quality of the elements. The latter point is a topic of current interest in many application areas where the element quality may degrade during computation as a result of repeated refinement or grid deformation. The target applications considered here are Lagrangian and Eulerian impact calculations since they represent a demanding class of problems that involve both issues – the need to adaptively refine the mesh and the presence of deforming elements. Another point of interest here is the need to provide appropriate error indicators that can be used in conjunction with *legacy* or *proprietary code* where access at the level of the physics or element detail may be limited. For this reason, we chose to essentially focus on error indicators that have this property while using the research version codes EPIC and CTH. The work on error and shape control includes some exploratory studies of hybrid indicators of additive or multiplicative type.

In this paper we have discussed several simple error indicators and adaptive strategies. Our preliminary results show that adapted meshes which improve the performance of the simulations have been obtained. In particular, the average aspect ratio, which is a simple measure of the mesh quality, is much improved. We have also established a sound technique for quantifying the errors due to mesh distortion using basic approximation theory and used it both singly and in combination with other estimators.

However, much remains to be done to use these indicators and strategies on realistic impact simulations. Principal among the tasks that need to be accomplished are the design of a dynamic data structure capable of supporting creation and deletion of arbitrary sets of elements, the implementation of mesh smoothing and local remeshing algorithms to augment element refinement and coarsening, the development of more reliable error estimators that account for the complex material behavior, a study of the interaction between adaptivity and different contact algorithms, and finally, the integration of mesh refinement, remeshing and smoothing with various error indicators into an adequate choice of techniques based on the different indicators.

Acknowledgements

The authors gratefully acknowledge support of this work through Contract No. DAHC94-96-C-0002 with the DoD High Performance Computing Modernization Program Major Shared Resource Center at the Engineer Research and Development Center in Vicksburg, Mississipi, and of support through ASCI II Grant No. B347883. In addition, we thank Drs. Richard Weed, Raju Namburu, Gordon Johnsson, Louis Turcotte and Joe Thompson for their suggestions and encouragement.

References

[1] M. Ainsworth and J. T. Oden. A posteriori error estimation in finite element analysis. *Comput. Methods Appl. Mech. and Engrg.*, 142:1–88, 1997.

[2] C. E. Anderson Jr. and S. R. Bodner. Ballistic impact: The status of analytical and numerical modelling. *Int. J. Impact Engrg.*, 16:9–35, 1988.

[3] I. Babuška and A. Miller. The post-processing approach in the finite element method - Part 3: A posteriori error estimates and adaptive mesh selection. *Internat. J. Numer. Methods Engrg.*, 20:2311–2324, 1984.

[4] I. Babuška and A. D. Miller. A feedback finite element method with a posteriori error estimation. Part I. *Comput. Methods Appl. Mech. and Engrg.*, 61:1–40, 1987.

[5] I. Babuška and W. C. Rheinboldt. Analysis of optimal finite element meshes in r^1. *Math. Comp.*, 33:435–463, 1979.

[6] I. Babuška, T. Strouboulis, C. S. Upadhyay, and S. K. Gangaraj. A model study of element residual estimators for linear elliptic problems: The quality of the estimators in the interior of meshes of triangles and quadrilaterals. *Computers & Structures*, 57(6):1009–1028, 1995.

[7] R. E. Bank and A. Weiser. Some a posteriori error estimators for elliptic partial differential equations. *Math. Comp.*, 44(170):283–301, 1985.

[8] R. C. Batra and K. I. Ko. An adaptive mesh refinement technique for the analysis of shear bands in plain strain compression of a thermoviscoplastic solid. *Comp. Mech.*, 10:369–379, 1992.

[9] E. B. Becker, G. F. Carey, and J. T. Oden. *Finite Elements – An Introduction.* Prentice-Hall, Englewood Cliffs, 1981.

[10] R. Becker and R. Rannacher. A feedback approach to error control in finite element methods: basic analysis and examples. *East-West J. Numerical Mathematics*, 4(4):237–264, 1996.

[11] D. J. Benson. Computational methods in Lagrangian and Eulerian hydrocodes. *Comput. Methods Appl. Mech. and Engrg.*, 99:235–394, 1992.

[12] J. U. Brackbill and J. S. Saltzman. Adaptive zoning for singular problems in two dimensions. *J. Comput. Phys.*, 46(3):342–368, 1982.

[13] D. Braess. *Finite Elements.* Cambridge University Press, Cambridge, UK, 1997.

[14] G. T. Camacho and M. Ortiz. Adaptive Lagrangian modelling of ballistic penetration of metallic targets. Preprint.

[15] G. F. Carey. *Computational Grids: Generation, Refinement, and Solution Strategies.* Taylor and Francis, Washington DC, 1997.

[16] G. F. Carey and A. Plaza. Local refinement of simplicial grids based on the skeleton. *Appl. Numer. Math.*, 32:195–218, 2000.

[17] D. Crawford. Adaptive mesh refinement in CTH. In *Proceedings of the U.S. Army Symposium on Solid Mechanics*, Myrtle Beach, SC, April 11-14 1999.

[18] E. Eriksson and C. Johnson. Adaptive finite-element methods for parabolic problems. Part I: A linear model problem. *SIAM J. Numer. Anal.*, 28(1):43–77, 1991.

[19] H. D. Espinosa, P. D. Zavattieri, and G. L. Emore. Adaptive FEM computation of geometric and material non-linearities with application to brittle failure. Preprint.

[20] P. L. George. *Automatic Mesh Generation.* Wiley, 1991.

[21] V. Girault and P.-A. Raviart. *Finite Element Methods for Navier-Stokes Equations.* Springer-Verlag, Berlin, 1986.

[22] A. Huerta, P. Díez, A. Rodríguez-Ferran, and G. Pijaudier-Cabot. Error estimation and adaptive finite element analysis of softening solids. In P. Ladevèze and J. T. Oden, editors, *Advances in Adaptive Computational Methods in Mechanics*, pages 333–347. Elsevier, Amsterdam, 1998.

[23] G. R. Johnson and R. A. Stryk. *User Instructions for the EPIC-3 code.* Air Force Armament Laboratory.

[24] G. R. Johnson and R. A. Stryk. Eroding interface and improved tetrahedral element algorithms for high-velocity impact computations in three dimensions. *Internat. J. Impact Engrg.*, 5:411–421, 1987.

[25] P. M. Knupp. Part II - A framework for volume mesh optimization and the condition number of the Jacobian matrix. Manuscript.

[26] P. M. Knupp and S. Steinberg. *The Fundamentals of Grid Generation.* CRC Press, Boca Raton, 1993.

[27] P. Ladevèze and D. Leguillon. Error estimate procedure in the finite element method and applications. *SIAM J. Numer. Anal.*, 20:485–509, 1983.

[28] P. Ladevèze and E. A. W. Maunder. A general method for recovering equilibrating element tractions. *Comput. Methods Appl. Mech. and Engrg.*, 137:111–151, 1996.

[29] P. Ladevèze and J. T. Oden, editors. *Advances in Adaptive Computational Methods in Mechanics*, number 47 in Studies in Applied Mechanics. Elsevier, Amsterdam, 1998.

[30] D. L. Littlefield and J. T. Oden. Implementation of adaptive mesh refinement in an Eulerian hydrocode. ERDC MSRC PET Technical Report No. TR00-04, Vicksburg, MS, 1999.

[31] D. L. Littlefield, J. T. Oden, and G. F. Carey. Simulation of hypervelocity impact using a block-adaptive, Eulerian impact mechanics code. In *Proceedings of the Advanced Simulation Technology Conference (ASTC 2000)*, Washington DC, April 16-20 2000.

[32] A. C. Mueller and G. F. Carey. Continuously deforming finite elements. *Internat. J. Numer. Methods Engrg.*, 21(11):2099–2126, 1985.

[33] Q. Niu and M. S. Shephard. Superconvergent boundary stress extraction and some experiments with adaptive pointwise error control. *Internat. J. Numer. Methods Engrg.*, 37:877–891, 1994.

[34] J. T. Oden and G. F. Carey. *Finite Elements - Mathematical Aspects, Vol. IV*. Prentice-Hall, Englewood Cliffs, 1983.

[35] J. T. Oden and A. Patra. A parallel adaptive strategy for *hp* finite element computations. *Comput. Methods Appl. Mech. and Engrg.*, 121:449–470, 1995.

[36] M. Ortiz and I. Quigley. Adaptive mesh refinement in strain localization problems. *Comp. Meth. App. Mech.*, 90:781–804, 1991.

[37] M. Paraschivoiu and A. T. Patera. A hierarchical duality approach to bounds for the outputs of partial differential equations. *Comput. Methods Appl. Mech. and Engrg.*, 158:389–407, 1998.

[38] M. Paraschivoiu, J. Peraire, and A. T. Patera. A posteriori finite element bounds for linear-functional outputs of elliptic partial differential equations. *Comput. Methods Appl. Mech. and Engrg.*, 150:289–312, 1997.

[39] A. Pardhanani and G. F. Carey. Optimization of computational grids. *Numer. Meth. Partial Differential Eqs.*, 4(2):95–117, 1988.

[40] S. Prudhomme and J. T. Oden. Residual error estimates for an explicit time discretization of the diffusion equation. In preparation.

[41] S. Prudhomme and J. T. Oden. On goal-oriented error estimation for elliptic problems: Application to the control of pointwise errors. *Comput. Methods Appl. Mech. and Engrg.*, 176:313–331, 1999.

[42] R. Rannacher and F. T. Suttmeier. A feedback approach to error control in finite element methods: Application to linear elasticity. Preprint 96-42, IWR, Heidelberg, 1996.

[43] R. Rodriguez. Some remarks on Zienkiewicz-Zhu estimator. *Numer. Meth. Partial Differential Eqs.*, 10:625–635, 1994.

[44] M. S. Shephard and P. M. Finnigan. Toward automatic model generation. In A. K. Noor and J. T. Oden, editors, *State-of-the-Art Surveys on Computational Mechanics*, volume 11, pages 335–366. ASME, 1991.

[45] M. Stynes. On faster convergence of the bisection method for all triangles. *Math. Comp.*, 35:1195–1201, 1980.

[46] J. F. Thompson. *Numerical Grid Generation.* Elsevier Science, New York, 1982.

[47] R. Verfürth. *A Review of A Posteriori Error Estimation and Adaptive Mesh-refinement Techniques.* Wiley-Teubner, Stuttgart, 1996.

[48] L. Wahlbin. *Superconvergence in Galerkin Finite Element Methods*, volume 1605 of *Lecture Notes in Mathematics.* Springer-Verlag, Berlin, 1995.

[49] S. Wang, I. H. Sloan, and D. W. Kelly. Computable error bounds for pointwise derivatives of a Neumann problem. *IMA J. Numer. Anal.*, 18:251–271, 1998.

[50] J. H. C. Whitehead. On C^1-complexes. *Ann. Math.*, 41:809–824, 1940.

[51] O. C. Zienkewicz and J. G. Zhu. A simple error estimator and adaptive strategy for practical engineering analysis. *Internat. J. Numer. Methods Engrg.*, 24:337–357, 1987.

[52] O. C. Zienkiewicz and J. Z. Zhu. The superconvergent patch recovery and a posteriori error estimates. Part 1: the recovery technique. *Internat. J. Numer. Methods Engrg.*, 33:1331–1364, 1992.

[53] O. C. Zienkiewicz and J. Z. Zhu. The superconvergent patch recovery and a posteriori error estimates. Part 2: error estimates and adaptivity. *Internat. J. Numer. Methods Engrg.*, 33:1365–1382, 1992.